Using run-time randomization against memory corruption attacks on legacy binaries

Chen, X.

2017

document version
Publisher's PDF, also known as Version of record

Samenvatting

Memory corruption vulnerabilities staan nog steeds in de top drie van de CWE SANS meest gevaarlijke software fouten sinds ze in de jaren zeventig zijn geïdentificeerd door Anderson. Zelfs nu zijn bijna 400 memory corruption fouten ontdenkt in CVEs in de afgelopen 10 jaar. Memory corruption fouten komen vooral voor in applicaties geschreven in type-unsafe talen zoals C en C++ vanwege de pointer operations ondersteund in deze talen. De mogelijkheid om pointers te beïnvloeden creëert een efficiënte manier om applicaties en systemen te ontwerpen, maar bij onjuist gebruik kan dit leiden tot onbedoelde veranderingen van aanverwante memory content, waardoor een aanvaller in staan is zo'n zwakheid te misbruiken door gevoelige data te compromitteren of de control flow over te nemen.
Het is moeilijk om ontwikkelaars over te laten stappen naar type-safe talen, niet alleen vanwege prestaties maar ook vanwege backward compatibility redenen. Het probleem met het verbeteren van de veiligheid van applicaties en systemen geschreven in C/C++ talen is al lang onderkend door de gemeenschap maar eerdere oplossingen gingen vaak gepaard met niet acceptable overheads en veel vereisen bron informatie die vaak niet beschikbaar is voor legacy binaries.

In dit proefschrift stellen we dat address space runtime randomization effectief kan zijn tegen memory corruption aanvallen met algemene toepassing mogelijkheden voor verschillende memory spaces: heap space, stack space en code space. Ons onderzoek focust zich op verbetering van de beveiliging garanties verstrekt door ASLR en eerder werk in dit gebied, en daarbij verbetering van performance, capabilities en re-randomization frequency. Om onze claims te bekrachtigen presenteren we meerdere contributies en demonstreren de levensvatbaarheid van de voorgestelde technieken

in de praktijk. In het bijzonder presenteren we de eerste onafhankelijke oplossing om legacy binaries te randomiseren bij runtime zonder noodzaak voor enig hardware, kernel of source code support.